**ФИО: Попов Кирилл Михайлович**

**Группа: СКБ252**

**1. Задание:**  
Разработать программу «Трекер книг» для учета прочитанной литературы. Программа должна позволять:

* Добавлять книги с информацией: название, автор, год прочтения, оценка (0-10)
* Просматривать список всех книг
* Выводить статистику: общее количество книг, средний рейтинг
* Обеспечивать корректный выход из программы

**2. Описание алгоритма:**

* Использование структуры Book для хранения данных о книге
* Главное меню с циклом while для постоянного взаимодействия с пользователем
* Функция loadBook () для загрузки записей файла
* Функция saveBookToFile() для сохранения добавленной книги
* Функция addBook() для добавления новой книги:
  + Ввод данных с клавиатуры
  + Проверка корректности оценки
  + Добавление книги в вектор
* Функция showAllBooks() для отображения всех книг в табличном формате
* Функция showStats () для расчета и вывода статистики:
  + Подсчет общего количества книг
  + Расчет среднего рейтинга
* Функция exitProgram() для корректного завершения работы

**3. Код:**

#include <iostream> // Для ввода/вывода через консоль (cin, cout)

#include <fstream>  // Для работы с файлами (ifstream, ofstream)

#include <cstring>  // Для работы со строками (char[])

#include <iomanip>  // Для красивого форматирования вывода (setw, right)

#include <windows.h>

using namespace std; // чтобы не писать каждый раз перед cin и cout

const int MAX\_BOOK = 1000;

struct Book

{

    char name[100];

    char author[50];

    int year;

    float grade;

};

Book books[MAX\_BOOK];

int book\_count = 0;

// Для загрузки записей файла

void loadBook(const char \*filename)

{

    ifstream fin(filename);

    if(!fin)

        return;

    while(fin >> books[book\_count].name >> books[book\_count].author >> books[book\_count].year >> books[book\_count].grade)

    {

        book\_count++;

    }

    fin.close();

}

// Сохранение добавленной книги

void saveBookToFile(const char \*filename, Book b)

{

    ofstream fout(filename, ios::app);

    if (!fout)

    {

        cout << "Ошибка при сохранении файла!\n";

        return;

    }

    fout << b.name << " " << b.author << " " << b.year << " " << b.grade << endl;

    fout.close();

}

// Добавить книгу

void addBook()

{

    if (book\_count >= MAX\_BOOK)

    {

        cout << "Превышен лимит количества книг.\n";

        return;

    }

    cin.ignore();

    Book b;

    cout << "Введите названия книги(используйте \_ вместо пробелов): ";

    cin.getline(b.name, 100);

    cout << "Введите автора: ";

    cin.getline(b.author, 50);

    cout << "Год прочтения: ";

    cin >> b.year;

    cout << "Ваша оценка (0-10): ";

    cin >> b.grade;

    books[book\_count++] = b;

    saveBookToFile("book.txt", b);

    cout << "Книга добавлена!\n";

    cin.ignore();

}

// Для показа всех книг

void showAllBooks()

{

    if (book\_count == 0)

    {

        cout << "Нет данных о книгах.\n";

        return;

    }

    cout << "\n--------- Все прочитанные книги ---------\n";

    cout << left << setw(70) << "Назавание произведения"

        <<  setw(40) << "Автор произведения"

        << setw(10) << "Год"

        << "Оценка(0-10)\n";

    cout << "------------------------------------------\n";

    for (int i = 0; i < book\_count; i++)

    {

            cout << left << setw(70) << books[i].name

            <<  setw(40) << books[i].author

            <<  setw(10) << books[i].year

            << books[i].grade << "\n";

    }

}

void showStats()

{

    if (book\_count == 0)

    {

        cout << "Нет данных для анализа.\n";

        return;

    }

    float total = 0;

    float average;

    float max = books[0].grade;

    char maxBook[100];

    strcpy(maxBook, books[0].name);

    for (int i = 0; i < book\_count; i++)

    {

        total += books[i].grade;

        if (books[i].grade > max)

        {

            max = books[i].grade;

            strcpy(maxBook, books[i].name);

        }

    }

    average = total / book\_count;

    cout << "Количество прочитанных книг: " << book\_count << "\n";

    cout << "Средняя оценка: " << average << "\n";

    cout << "Самая высокая оценённая книга - " << maxBook << ": " << max << "\n";

}

void showMenu()

{

    cout << "\n=== Трекер книг ===\n";

    cout << "1. Добавить книгу\n";

    cout << "2. Показать все книги\n";

    cout << "3. Показать статистику\n";

    cout << "4. Выход\n";

    cout << "Выберите пункт: ";

}

int main()

{

    // Переключаем консоль в UTF-8

    SetConsoleOutputCP(65001); // 65001 = UTF-8

    SetConsoleCP(65001);

    loadBook("book.txt");

    int choice; // переменная для хранения выбора

    do{

        showMenu();

        cin >> choice;

        switch(choice)

        {

        case 1:

            addBook();

            break;

        case 2:

            showAllBooks();

            break;

        case 3:

            showStats();

            break;

        case 4:

            cout << "Выход из программы.\n";

            break;

        default:

            cout << "Неверный выбор.\n";

        }

    }while(choice != 4);

    return 0;

}

**4. Ссылка на репозиторий:** <https://github.com/P0KI-7/Trecker-book>

**5. Вывод:**  
Разработал программу "Трекер книг" на C++. Разобрался с работой со структурами, векторами, форматированным выводом. Реализовал интерактивное меню и основные функции для управления коллекцией книг. Программа удобна для учета прочитанной литературы и анализа reading-привычек.